UNA TÉCNICA DE AGRUPACIÓN ROBUSTA PARA UN ENFOQUE BIG DATA: CLARABD PARA TIPOS DE DATOS MIXTOS

A robust clustering technique for a Big Data approach: CLARABD for Mixed data types

[1,2,3]Víctor Morales-Oñate, [4]Bolívar Morales-Oñate

[1]Instituto de Estadística, Universidad de Valparaíso, Valparaíso, Chile
[2]Departamento de Desarrollo, Ambiente y Territorio, Facultad Latinoamericana de Ciencias Sociales, Quito, Ecuador
[3]Analítica de Datos, IDCE Consulting, Quito, Ecuador
[4]Facultad de Ciencias, Escuela Superior Politécnica de Chimborazo, Riobamba - Ecuador.

*bolivar.morales@espoch.edu.ec

Resumen

Cuando el investigador no cuenta con un conocimiento apriori de la conformación de grupos en un conjunto de datos dado, emerge la necesidad de realizar una clasificación conocida como clasificación no supervisada. Además, el conjunto de datos puede ser mixto (datos cualitativos y/o cuan- titativos) o presentarse en grandes volúmenes. El algoritmo k-medias, por ejemplo, no permite la comparación de datos mixtos y está limitado a un máximo de 65536 objetos en el software R. K-medoides, por su parte, permite la comparación de datos mixtos pero también tiene la misma limitación de objetos que k-medias. El algoritmo CLARA tradicional puede exceder fácilmente este limitante de volúmenes, pero no permite la comparación de datos mixtos. En este contexto, este trabajo es una extensión del algoritmo CLARA para datos mixtos, el algoritmo CLARABD. La distancia de Gower es central en CLARABD para realizar esta extensión, debido a que permite la comparación de datos mixtos y también es posible procesar un conjunto de datos con mas de 65536 observaciones. Para mostrar las bondades del algoritmo propuesto, se ha realizado un proceso de simulación así como una aplicación a datos reales obteniendo resultados consistentes en cada caso.

Palabras clave: Clasificacion, CLARA, K-medoides, datos mixtos, R software.

Abstract

When a researcher does not have an a priori knowledge of the configuration of groups in a given data set, the need to perform a classification known as unsupervised classification emerges. In addition, the data set can be mixed (qualitative and/or quantitative data) or presented in large volumes. The kmeans algorithm, for example, does not allow the comparison of mixed data and is limited to a maximum of 65536 objects in the R software. K-medoids, on the other hand, allows the comparison of mixed data but also has the same limitation of objects that k-means does. The traditional CLARA algorithm can easily exceed this volume limitation, but it does not allow the comparison of mixed data. In this context, this work is an extension of the CLARA algorithm for mixed data, the CLARABD algorithm. Gower distance is central in CLARABD to make this extension, because it allows the comparison of mixed data and it is also possible to process a data set with more than 65536 observations. To show the benefits of the proposed algorithm, a simulation process has been carried out as well as an application to real data, obtaining consistent results in each case.

Palabras clave: Classification, CLARA, K medoids, mixed data types, R software.

Fecha de recepción: 11-01-2019 **Fecha de aceptación:** 11-06-2019

I. INTRODUCCION

La búsqueda de grupos en un conjunto de datos no es una tarea nueva. Uno de los libros seminales a este respecto fue publicado por primera vez en 1990, Finding groups in data: An introduction to *cluster analysis* [1]. Lo que en ese entonces se conoce en inglés como *cluster analysis* se traduce al español como análisis de *conglomerados*. Comúnmente se podía encontrar estos métodos en trabajos relacionados a estadística multivariante [2]. Hoy, con el surgimiento de la ciencia de datos y del big data [3], el análisis de conglomerados se conoce como *clasificación no supervisada o aprendizaje no supervisado*, llevando este nombre debido a que el investigador no conoce apriori, la clase o grupo al que pertenecen las observaciones del conjunto de datos que analiza.

La investigación teórica y aplicada que usa algoritmos de clasificación no supervisada sigue vibrante. Por ejemplo, existen trabajos en diferentes áreas como clasificación de imágenes [4], analítica de deportes [5], análisis de lenguaje [6] y ciencias sociales [7] en esta línea de investigación. Cada uno usa diferentes métodos con un mismo propósito: encontrar grupos en los datos analizados. Dos de los algoritmos clásicos con los que se ha abordado este problema son k-medias y k-medoides. Ambos tienen como entrada el conjunto de datos y el número de conglomerados, como salida una partición del conjunto. K-medias permite usar únicamente variables cuantitativas y kmedoides permite usar variables cuantitativas y cualitativas. Una extensión de este ´ultimo es el algoritmo CLARA [1].

CLARA es conocida por ser una alternativa robusta para clasificación no supervisada para conjuntos de datos *grandes*. Por un lado, se considera robusta por usar el algoritmo k-medoides para obtener los representantes de los grupos. Por otro lado, un conjunto de datos es considerado grande en función de la complejidad computacional así como del poder de cómputo. Por ejemplo, en 1990 se entendía como un conjunto de datos *grande* cuando se tenía más de 100 observaciones [1].

Hoy se considera *grande* un problema con varios miles de observaciones. En particular, el software R permite hasta 65536 observaciones cuando usa el algoritmo k-medoides tradicional; más allá de ese umbral el problema debe ser abordado con el algoritmo CLARA [8]. En el contexto Big Data, CLARA es un algoritmo que encaja adecuadamente. Es capaz de procesar conjuntos de datos de millones de observaciones en pocos segundos [9]. Sin embargo, una limitante es la métrica utilizada para el cálculo de las disimilaridades (diferencia o distancia). Actualmente las opciones son la distancia euclideana y la de manhattan. Esto limita el uso de este potente algoritmo para la clasificación de datos mixtos, esto es, datos de tipo nominal, ordinal y binario (a) simétricos.

El algoritmo CLARA realiza múltiples muestras del conjunto de datos original, aplica k-medoides a cada muestra, encuentra los medoides y luego devuelve su mejor agrupamiento como salida.

Este trabajo presenta el algoritmo CLARABD que extiende al algoritmo tradicional posibilitando la clasificación de observaciones con tipos de datos mixtos y ha sido implementado en el lenguaje R [10]. Específicamente, esta propuesta se diferencia del algoritmo CLARA tradicional en que la entrada para el cálculo de los medoides de cada muestra puede realizarse mediante una matriz de distancias o disimilaridad con las métricas euclídea, manhattan y gower. Siendo esta última la métrica de disimilaridad la que permite clasificar observaciones de tipo mixto.

Marco Teórico

Es común encontrar definiciones de clustering en la literatura de análisis multivariante, machine learning y reconocimiento de patrones. Se cita a continuación tres definiciones:

- Todo se relaciona con la agrupación o segmentación de una colección de objetos en subconjuntos o clúster, de modo que aquellos dentro de cada clúster están más estrechamente relacionados entre sí que los objetos asignados a diferentes clúster. [11]
- El clustering se refiere a un conjunto muy amplio de técnicas para encontrar subgrupos, o clústeres, en un conjunto de datos. Cuando se agrupan las ob-

servaciones de un conjunto de datos, se busca dividirlas en grupos distintos para que las observaciones dentro de cada grupo sean bastante similares entre sí, mientras que las observaciones en diferentes grupos son bastante diferentes entre sí. [12]

• El análisis cluster, que es el ejemplo más conocido de aprendizaje no supervisado, es una herramienta muy popular para analizar datos multivariados no estructurados. Dentro de la comunidad de minería de datos, el análisis de clúster también se conoce como segmentación de datos, y dentro de la comunidad de aprendizaje automático también se conoce como descubrimiento de clases. La metodología consiste en varios algoritmos, cada uno de los cuales busca organizar un conjunto de datos determinado en subgrupos homogéneos, o clúster. [13]

Sin embargo, es menos común encontrar un marco unificador para el problema de clasificación. En este sentido, [14] presenta un marco que permite esta conexión. Su propuesta se denomina Modelo Estructural de Cubrimientos (MEC) y ha motivado trabajos posteriores como [15] y [16]. A continuación se revisa brevemente los ítems más relevantes de su teoría.

Definicion 1 (Estructura-Cubrimiento). Una estructura-cubrimiento es una tupla con los siguientes elementos: (Ω, $\Re$, δ, Q, π, f), donde

• Ω es un conjunto no vacío de objetos $\Omega = \{o_1, o_2, \ldots, o_n\}$.
• $\Re$ es un conjunto $\Re = \{x_1, x_2, \ldots, x_r\}$ de variables llamadas Rasgos descriptivos, en función de los cuales se pueden describir los objetos en Ω. Cada rasgo descriptivo xi tiene un Dominio de definición M_i. Cada rasgo x_i tiene un dominio D_i.
• δ es una relación funcional $\delta : \Omega \rightarrow (D_1 \times D_2 \times \ldots \times D_r)$ llamada Relación de Descripción que a cada objeto le asigna una descripción en términos de los rasgos en $\Re$.
• Q Es un conjunto $Q = \{C_1, C_2, \ldots, C_k\}$, de etiquetas correspondientes a conjuntos llamados Clases o Categorías, en los cuales se agrupan los elementos de Ω.
• π es una relación funcional $\pi : (\Omega \times Q) \rightarrow [0, 1]$ llamada Relacion de Pertenencia o Función de Pertenencia que a cada pareja, (objeto, clase), le asocia un grado de pertenencia.
• f es una relación funcional llamada Función de Analogía entre Patrones. Es una función de comparación entre patrones que puede ser de semejanza o diferencia.

La estructura-cubrimiento permite formalizar los diferentes problemas de clasificación que el investigador puede enfrentar. En cada tipo de problema se recibe como datos iniciales algunos elementos de una estructura-cubrimiento, y se requiere encontrar los elementos faltantes. Si el problema es supervisado, se conoce la familia de clases y algunos elementos de pertenencia. Esto es, se tiene Ω, $\Re$, δ, Q, π (π se conoce parcialmente), pero no se dispone de π ni f. Si, por el contrario, el problema es no-supervisado, entonces, no se conoce más que los objetos y sus descripciones. Es decir, se tiene Ω, $\Re$, δ, pero no Q, π ni f.

El único elemento que está siempre ausente en todo problema de clasificación y que se convierte en el objetivo del método, es determinar es la función de comparación entre patrones. La selección de la función de comparación es la decisión más importante en el proceso de solución de un problema. En esa selección influyen la experiencia del modelador y las recomendaciones del experto.

Es adecuado mencionar también algunos tipos de cubrimiento:
• Disjunto: Todas las clases son disjuntas.
• Solapado: Algunas clases se intersectan.
• Total: Todos los objetos pertenecen a a alguna clase.
• Parcial: No todos los objetos pertenecen a alguna clase.

Note que la función de comparación (semejanza o diferencia) puede considerarse como la piedra angular del MEC. En este sentido, existen diferentes formas de comparar objetos (también llamados patrones).

Definición 2 (Función de Comparación). Sean A y B dos objetos en el espacio E,

$A = (a_1, a_2, \ldots, a_r)$

$B = (b_1, b_2, \ldots, b_r)$

entonces f(A, B) es una función de comparación de objetos tal que,
$f : E \times E \rightarrow \Gamma$.

Donde Γ es un conjunto totalmente ordenado

A partir de la definición 2, se derivan dos tipos de funciones de comparación: de semejanza y diferencia. Una función de semejanza, donde el conjunto de salida es [0, 1], mide el grado de acuerdo, coincidencia o relación, entre dos objetos, a partir del valor de cada uno de sus atributos. Una función de diferencia, donde el conjunto de llegada es [0, ∞), mide el concepto opuesto, es decir, el grado de desacuerdo o incompatibilidad entre dos objetos.

En la propuesta del algoritmo CLARABD de este trabajo, se usan funciones de distancia. Particularmente, las distancias euclideana, manhattan y gower:

• Euclideana: es la distancia más usada y conocida, su fórmula de cálculo es:

$$d_e(A, B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

• Manhattan: es usada generalmente en espacios discretos (posiblemente infinitos), su fórmula de cálculo es:

$$d_{Manhattan}(A, B) = \sum_{i=1}^{n} |a_i - b_i|$$

• Gower: usada cuando se dispone de tipos de datos mixtos (cualitativos y cuantitativos). Su forma de cálculo es:

$$d_{gower}(A, B) = \sqrt{1 - s(A, B)}$$

donde s(A, B) es el coeficiente de similaridad de Gower:

$$s(A, B) = \frac{\sum_{h=1}^{p_1}(1 - |a_{ih} - b_{ih}|/G_h) + r + \alpha}{p_1 + (p_2 - d) + p_3}$$

y p_1 es el número de variables cuantitativas continuas, p_2 es el número de variables binarias, p_3 es el número de variables cualitativas (no binarias), r es el número de coincidencias (1, 1) en las variables binarias, d es el número de coincidencias (0, 0) en las variables binarias, α es el número de coincidencias en las variables cualitativas (no binarias) y G_h es el rango (o recorrido) de la h-esima variable cuantitativa [17].

Teniendo en cuenta estos elementos, se puede definir un algoritmo de clasificación.

Definición 3 (Algoritmo de clasificación). Un Algoritmo de Clasificación, A, recibe como entrada un cubrimiento parcial y lo transforma en un cubrimiento total.

$A[(\Omega, \Re, \delta, Q_1, \pi_1, f)] = (\Omega, \Re, \delta, Q_2, \pi_2 f)$

Note que en el cubrimiento final pueden aparecer modificados tanto el conjunto de clases como la relación de pertenencia. Además, en todo proceso de clasificación se debe tener en cuenta su principio fundamental: objetos semejantes pertenecen a la misma clase; objetos diferentes pertenecen a clases distintas [14].

A partir del principio, es claro que lo deseable es que la semejanza dentro del grupo sea lo más alta posible. Es decir, los grupos resultantes de un algo- ritmo de clasificación deberían tener mayor semejanza interior que exterior. La varianza, que clásicamente es usada en estadística, recoge la noción de semejanza interior. Es decir, se desea obtener grupos con la menor varianza posible.

Otro elemento importante al momento de comparar objetos, es considerar el caso en el que los objetos pertenezcan a un espacio diferente de $\mathbb{R}^n$. Por ejemplo, si se tiene los objetos O_1 = (dulce, 27, sábado, grande) y O_2 = (salado, 25, jueves, mediano), no es posible usar la distancia euclideana porque los espacios contienen valores categóricos.

Note que la distancia de Gower si podría abordar el problema, pero hay que tener en cuenta ciertos elementos. A continuación se usa este ejemplo para ilustrar elementos de la definición 1. En este caso, los objetos O_1 y O_2 tienen a sabor, edad, día y tamaño como rasgos descriptivos. Se sabe que cada rasgo

descriptivo tiene su propio dominio, en este caso serían
• Dom(sabor) = {dulce, salado, agrio, ácido}
• Dom(edad) = [0, 120] enteros
• Dom(día) = {lunes, martes, miércoles, jueves, viernes, sábado, domingo}
• Dom(tamaño) = {pequeño, mediano, grande}

Si bien la distancia eculideana no puede ser usada en este caso, una forma general de abordar este problema es notar que los objetos O_1 y O_2 están en el mismo espacio. Se puede proceder a compararlos en cada rasgo usando funciones auxiliares.

En este ejemplo se pueden definir cuatro funciones auxiliares, una para cada rasgo: g_1(dulce, salado), g_2(27, 25), g_3(-sábado, jueves) y g_4(grande, mediano). Si todas las funciones $g_i(x, y)$, i = 1, 2, 3, 4 son de diferencia, entonces se tiene diferencias parciales respecto a cada rasgo descriptivo entre los objetos. Ahora, para obtener una medida de diferencia global, se deben combinar de alguna manera las diferencias parciales. Específicamente, se puede usar la función de distancia sintáctica.

Definición 4 (Distancia y semejanza sintáctica). Sean A = (a_1, a_2, . . . , a_r) y B = (b_1, b_2, . . . , b_r) dos objetos y $g_i(x, y)$, i = 1, . . . , r, funciones auxiliares, la distancia sintáctica se define como

$$D_s(A, B) = \sum_{i=1}^{r} \alpha_i [g_i(a_i, b_i)]$$

En el mismo contexto, la semejanza sintáctica se define como

$$S_s(A, B) = \frac{1}{r} \sum_{i=1}^{r} \alpha_i [g_i(a_i, b_i)]$$

donde $\sum_{i=1}^{r} \alpha_i = 1$ α_i=1 pondera la relevancia de cada rasgo.

La distancia sintáctica permite la comparación de objetos en cualquier espacio de representación e incluso ponderando la relevancia de cada rasgo con el requisito de que las funciones auxiliares est´en bien definidas. Note que la distancia de Gower ya permite trabajar con datos mixtos, pero clara- mente la aproximación al problema usando funciones auxiliares y distancias sintácticas es todavía más general.

II. MATERIALES Y MÉTODOS

Como se puede apreciar, el MEC ofrece un marco general para abordar el problema de clasificación. El algoritmo CLARABD de este trabajo es una extensión del algoritmo CLARA. Su desarrollo se muestra en esta sección y para ello se necesita presentar dos algoritmos que son su insumo: K-medias y K-medoides.

K-medias

Por primera vez desarrollado por [18], el algoritmo k-medias es quizá el Algoritmo de clasificación no jerárquica más utilizado en toda la literatura. Sea en textos de contenido teórico como [19, 20] o textos aplicados como [9, 21], siempre esta presente una sección dedicada al algoritmo k-medias. A continuación se presenta el algoritmo.

El algoritmo recibe como entrada a un conjunto de objetos en el espacio $\mathbb{R}^n$ y *k* (número de grupos a formar). El resultado una partición del espacio de objetos, tal que, optimiza la varianza global.

1. Calcular la Matriz Global de Distancias.
2. Seleccionar, los k objetos más alejados, como atractores iniciales.
3. Calcular y almacenar la distancia entre cada objeto y cada uno de los k atractores.
4. Particionar el espacio en grupos, asignando cada objeto al grupo del atractor más cercano.
5. Calcular, para cada grupo definido, su centroide.
6. Considerar los centroides recién calculados como nuevos puntos atractores.
7. Regresar al paso (3).
8. Terminar cuando el conjunto de centroides sea idéntico que el de la iteración anterior.

K-medoides

Cabe señalar el algoritmo k-medoides es un método de clasificación no supervisado. Un lector familiarizado con el algoritmo k-medias encontrará grandes similitudes. Siguiendo a [1], se presenta a continuación el algoritmo.

1. Seleccionar una función de comparación entre objetos. Por ejemplo, si se trata de variables cualitativas se suele usar la distancia euclideana, en este trabajo se usa la distancia de Gower.
2. Calcular la Matriz Global de semejanza/diferencia, esto es, la matriz de distancias.
3. Seleccionar, los *k* patrones más alejados, como atractores iniciales.
4. Calcular y almacenar la semejanza/diferencia entre cada patrón y cada uno de los *k* objetos atractores
5. Particionar el espacio en grupos, asignando cada patron al grupo del atractor más cercano
6. Calcular, para cada grupo definido, su medoide
7. Considerar los medoides reci´en calculados como nuevos patrones atractores.
8. Regresar al paso (4)
9. Terminar cuando el conjunto de medoides sea idéntico que el de la iteracion anterior.

La última partici´on obtenida, (idéntica a la de la iteración anterior) es la respuesta final del algoritmo. Con el algoritmo k-medoides se tiene un mecanismo para agrupar (por particionamiento) objetos en cualquier espacio de representación. Por el hecho de calcular medoides en lugar de centroides, el algoritmo k-medoides converge más rápido a la única solución global posible en ese espacio de representación y con ese conjunto de objetos.

CLARA
El algoritmo CLARA nace ante la necesidad de superar las barreras de memoria y tiempo de cómputo del algoritmo k-medoides (también conocido como *Partitioning Around Medoids* PAM) y está claramente explicado en el capítulo 3 de [1].

El método consiste, en términos generales, de dos pasos. Primero, se obtiene una muestra de objetos de los cuales se generan *k* grupos usando el algoritmo k-medoides. Es decir, se tienen *k* objetos representativos (medoi- des) de cada grupo. Segundo, cada objeto que no pertenece a la muestra es asignado al objeto más cercano de los *k* representativos. Esto resulta en una partición de todo el conjunto de objetos.

En el segundo paso se calcula la distancia promedio entre cada objeto de todos los datos y su objeto representativo. Después de realizar este proceso varias veces (el valor por defecto suele ser 5), se escoge la partición para la cual se tiene la distancia promedio más baja. En términos más específicos, los pasos del algoritmo CLARA son [22]:

1. Dividir aleatoriamente los conjuntos de datos en múltiples subconjuntos con tamaño fijo.
2. Calcular el algoritmo PAM en cada subconjunto y elegir los *k* objetos representativos correspondientes (medoides). Asignar cada observacion del conjunto de datos completo al medoide más cercano.
3. Calcular la media (o la suma) de las diferencias de las observaciones con su medoide más cercano. Esto se usa como una medida de la bondad de la agrupación.
4. Retenga el subconjunto de datos para el que la media (o suma) es mínima.

CLARABD
CLARABD es un algoritmo que tiene dos objetivos. En primer lugar, extiende el algoritmo CLARA tal que pueda usarse la distancia de Gower para obtener la agrupación final. Es decir, permite usar datos mixtos (datos de tipo nominal, ordinal y binarios (a)simétricos). K-medoides también puede usarse con datos mixtos. Sin embargo, en el programa R existen limitaciones en su cálculo. Actualmente existe un límite estricto, el nu´mero de objetos debe ser menor o igual a 65536. Cuando el número objetos supera este límite, se sugiere usar el algoritmo CLARA. En este sentido, el segundo punto en el que CLARABD extiende a k-medoides en R es porque permite realizar agrupaciones más allá de 65536 objetos.

Específicamente, el punto 2 del algoritmo CLARA es adaptado en CLA- RABD. Este paso del algoritmo en CLARABD sería.

Con la opción de elegir la distancia de Gower además de la euclídea y manhattan, se calcula el algoritmo PAM en cada subconjunto y se eligen los k objetos representativos correspondientes (medoides). Asignar cada observación del conjunto de datos completo al medoide

más cercano.

Es decir, todas las disimilaridades requeridas por el algoritmo pueden ser calculadas con la distancia de Gower. El código del algoritmo CLARABD ha sido plasmado en la función claraBD y puede descargarse de https://github.com/vmoprojs.

III. RESULTADOS

Simulacion

Para evaluar los resultados de CLARABD, se ha configurado el siguiente escenario de simulación. Se crearon 4 grupos de tamaño 100, un total de 400 objetos. Cada uno está compuesto de dos atributos, una variable cuantitativa y una nominal.

La variable cuantitativa fue construida generando números aleatorios que siguen una distribución normal con distintas medias para cada grupo y varianza constante. La variable categórica fue construida mediante la generación de números aleatorios con distribución binomial con probabilidad
0,5. El código para reproducir los resultados de la simulación se encuentra en el apéndice A.

La figura 1 muestra el ratio within/between para evaluar la consistencia respecto al número de grupos. Efectivamente, se observa que el ratio disminuye a medida que aumenta el número de grupos, lo que muestra consistencia en la agrupacion. También se muestran los valores para k-medoides como referencia.

Se puede apreciar el ratio dentro (within) y entre (between) la suma de cuadrados tiene una caída significativa en 4 grupos en CLARABD. K-medoides, en contraste, decrece más lentamente. Es posible que, al usar muestras de los datos originales, CLARABD tiene más probabilidad de capturar los cambios en estructura de la agrupación a medida que aumenta el número de grupos.

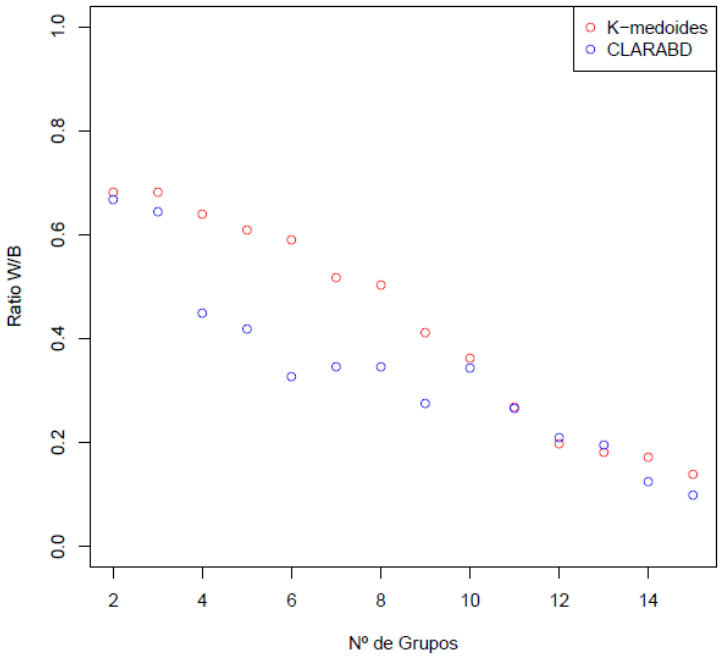

Figura 1: Ratio within/between vs nu´mero de grupos

Aplicación

A continuación se aplica el algoritmo CLARABD a un conjunto de datos de crédito de un banco alemán. Estos datos se obtuvieron del Repositorio de Aprendizaje Automático de la Universidad de California [23]. El conjunto de datos, que contiene atributos y resultados sobre 1000 solicitudes de préstamo, fue proporcionado en 1994 por el Profesor Dr. Hans Hofmann del Instituto de Estadística y Econometría de la Universidad de Hamburgo. Ha servido como un importante conjunto de datos de prueba para varios algoritmos de puntuación de crédito.

Una descripción más detallada de los datos puede encontrarse en el repositorio así como en la figura 5. Cuenta con 21 en total (incluyendo el identificador de cliente), variables cuantitativas (duraci´on del crédito, monto, edad, entre otros) y cualitativas (historial de crédito, destino del crédito, si es extranjero, entre otras).

Los datos contienen una variable de incumplimiento de crédito y, como un ejercicio de validación, se asume esta variable como determinada por las demás variables del conjunto de datos. Esto permite calcular una matriz de confusión usando CLARABD y k-medoides.

La tabla 1 muestra los porcentajes respecto al total de la matriz de con fusion. Su objetivo es mostrar que los porcentajes son parecidos en la clasificación. Los valores

sin paréntesis son los resultados de CLARABD y en paréntesis estan los resultados de k-medodides. El apéndice B. contiene el codigo que reproduce los resultados de la tabla 1.

Grupo	No incumplimiento	Incumplimiento
1	0, 517 (0, 569)	0, 215 (0, 208)
2	0, 183 (0, 131)	0, 085 (0, 092)

Tabla 1: Matriz de confusion. CLARABD y en paréntesis k-medoides

Otra forma de comparar los resultados de los algoritmos es a través de los representantes de los grupos y cuan separados se encuentran. Los medoides que resultan de CLARABD son los objetos 10 y 843, PAM tiene por medoides a 892 y 261.

La tabla 2 muestra estos resultados y se puede apreciar que en algunas variables coinciden y en otras no. Por ejemplo, en la variable ahorro se tiene exactamente el mismo resultado, mientras que en la variable monto la distancia es amplia entre los medoides. En general, existen diferencias en 12 de las 20 variables en ambos casos.

VARIABLES	CLARABD		PAM	
Estado	A12	A14	A14	A11
Duración	30	18	15	12
Historia	A34	A32	A34	A32
Propósito	A40	A45	A43	A42
Monto	5234	1943	1829	1657
Ahorros	A61	A61	A61	A61
Empleo	A71	A72	A75	A73
Instalado	4	4	4	2
Estado	A94	A92	A93	A93
Otro	A101	A101	A101	A101
Residencia	2	4	4	2
Propiedad	A123	A121	A123	A121
Edad	28	23	46	27
OtrosPlanes	A143	A143	A143	A143
Casa	A152	A152	A152	A152
Tarjetas	2	1	2	1
Trabajo	A174	A173	A173	A173
Confiable	1	1	1	1
Teléfono	A191	A191	A192	A191
Extranjero	A201	A201	A201	A201
Objeto	10	843	892	261

Tabla 2: Centroides de cada partición, CLARABD y PAM.

Finalmente, la figura 2 muestra un gráfico biplot de las variables numéricas. Este gráfico utiliza las dos componentes más relevantes como resultado de aplicar un análisis de componentes principales sobre las variables numéricas. Ambas componentes representan el 44 % del total de la varianza de los datos y se muestran las etiquetas de la partición encontrada por los algoritmos CLARABD y PAM. Se puede apreciar que los patrones de los conglomerados son similares.

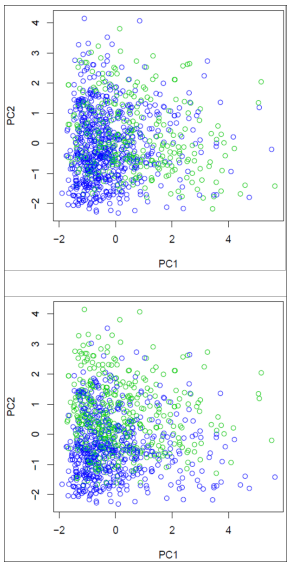

Figura 2: Biplot de variables numéricas etiquetadas por las agrupaciones de dos grupos resultantes. En el panel izquierdo presenta el algoritmo CLARBD, en el lado derecho se presenta el algoritmo PAM

```
rm(list = ls())
gc()
library(cluster)
library(fpc)

source("~/.../ClaraFunction.R")


#ST: Parameters:
metric <- "gower"
#END: Parameters:

## ST: generate objects, divided into 2 clusters.
k <- 4
samples = 5

n1 <- 100;m1 <- 0
n2 <- 100;m2 <- 5
n3 <- 100;m3 <- 10
n4 <- 100;m4 <- 15
n1+n2+n3+n4

nsam <- 15
SOLclstats <- array(NA,dim = c( 5,3,nsam))
SOLclus <- array(NA,dim = c( 4,2,nsam))
clstats <- NULL
for(sam in 2:nsam)
{
  x <- rbind(cbind(rnorm(n1,m1,0.5),
  rnorm(n1,m1,0.5)),
  cbind(rnorm(n2,m2,0.5),
  rnorm(n2,m2,0.5)),
  cbind(rnorm(n3,m3,0.5),
  rnorm(n3,m3,0.5)),
  cbind(rnorm(n4,m4,0.5),
  rnorm(n4,m4,0.5)))

  x <- as.data.frame(x)
  x$nominal1 <- factor(rbinom(nrow(x),k,0.5))
  x <- x[,c(1,3)]

  # PAM
  A <- pam(x,sam,metric = metric)
  #CLARABD
  C <- claraBD(x,sam,clus = FALSE,metric = metric)

  d <- daisy(x,metric = metric)
  A.validation <- cluster.stats(d,A$clustering)
  C.validation <- cluster.stats(d,C$clustering)

  WB <- c(A.validation$wb.ratio,
  C.validation$wb.ratio)
  WB <- c(WB,which.min(WB))


  clstats <- rbind(clstats,WB)
  colnames(clstats) <- c("ClaraBD",
  "Clara","Criterio")
}

pdf("Ratio.pdf")
plot(2:nsam,clstats[,1], xlab = "N°␣de␣Grupos"
     ,ylab = "Ratio␣W/B",col = "red", ylim = c(0,1))
points(2:nsam,clstats[,2], col = "blue")
legend("topright",c("K-medoides","CLARABD"),
pch = 1, col = c("red","blue"))
dev.off()
```

Figura 3: código de simulación

```
rm(list = ls())
gc()
library(cluster)
uu <-
"http://www.biz.uiowa.edu/faculty/
jledolter/DataMining/germancredit.csv"
credit <- read.csv(uu)

source("~/.../ClaraFunction.R")
```

```
metric <- "gower"
d <- daisy(credit,metric = metric)

paBDdefpam <- pam(credit[,-1], k = 2, metric = metric)
tab2 <- table(paBDdefpam$clustering,credit$Default)
sum(diag(prop.table(tab2)))

set.seed(89)# use this for reprodicibility
paBDdef <- claraBD(credit[,-1], k = 2, clus = FALSE, metric = metric)
tab1 <- table(paBDdef$clustering,credit$Default)

prop.table(tab1)
prop.table(tab2)
```

Figura 4: código de aplicación

Atributo 1: (cualitativo)
Estado de la cuenta corriente existente
A11: ... <0 DM
A12: 0 <= ... <200 DM
A13: ...>= 200 asignaciones de DM / salario por al menos un año
A14: no cuenta corriente

Atributo 2: (numérico)
Duración en mes

Atributo 3: (cualitativo)
Historial de crédito
A30: no se toman créditos / todos los créditos se devuelven debidamente
A31: todos los créditos en este banco pagados debidamente
A32: créditos existentes pagados hasta ahora.
A33: retraso en pagar en el pasado
A34: cuenta crítica / otros créditos existentes (no en este banco)

Atributo 4: (cualitativo)
Propósito
A40: coche (nuevo)
A41: coche (usado)
A42: mobiliario / equipamiento
A43: radio / televisión
A44: electrodomésticos
A45: reparaciones
A46: educación
A47: (vacaciones - no existe?)
A48: reentrenamiento
A49: negocios
A410: otros

Atributo 5: (numérico)
Monto de crédito

Attibute 6: (cualitativo)
Cuenta de ahorros / bonos
A61: ... <100 dm
A62: 100 <= ... <500 DM
A63: 500 <= ... <1000 DM
A64: ..>= 1000 DM
A65: desconocido / no cuenta de ahorros

Atributo 7: (cualitativo)
Empleo actual desde
A71: desempleados
A72: ... <1 año
A73: 1 <= ... <4 años
A74: 4 <= ... <7 años
A75: ..>= 7 años

Atributo 8: (numérico)
Tasa de entrega en porcentaje de la renta disponible

Atributo 9: (cualitativo)
Estado personal y sexo
A91: hombre: divorciado / separado
A92: mujer: divorciada / separada / casada
A93: hombre: soltero
A94: varón: casado / viudo
A95: femenino: soltero

Atributo 10: (cualitativo)
Otros deudores / garantes
A101: ninguna
A102: co-solicitante
A103: garante

Atributo 11: (numérico)
Residencia actual desde

Atributo 12: (cualitativo)
Propiedad
A121: inmobiliaria
A122: si no es A121: acuerdo de ahorro de la sociedad de construcción/seguro de vida
A123: si no es A121 / A122: automóvil u otro, no en el atributo 6
A124: desconocido / sin propiedad

Atributo 13: (numérico)
Edad en años

Atributo 14: (cualitativo)
Otros planes de cuotas
A141: banco
A142: tiendas
A143: ninguno

Atributo 15: (cualitativo)
Alojamiento
A151: alquiler
A152: propio
A153: gratis

Atributo 16: (numérico)
Número de créditos existentes en este banco.

Atributo 17: (cualitativo)
Trabajo
A171: desempleados / no calificados - no residentes
A172: no calificado - residente
A173: empleado calificado / oficial
A174: dirección / autónomos /
empleado / oficial altamente calificado

Atributo 18: (numérico)
Número de personas que son responsables de dar mantenimiento a

Atributo 19: (cualitativo)
Teléfono
A191: ninguno
A192: sí, registrado bajo el nombre de los clientes

Atributo 20: (cualitativo)
trabajador extranjero
A201: si
A202: no

Figura 5: Descripción de los atributos de la aplicación

III. CONCLUSIONES

Se ha logrado extender el algoritmo CLARA para datos mixtos. Para esto se usa la distancia de gower dentro del algoritmo CLARA tradicional. Los resultados de este proceso se han plasmado desde una perspectiva de simulación para evaluar la consistencia en la configuración de las agrupaciones finales así como una aplicación

a datos reales. Ambos enfoques comparan los resultados con k-medoides dado que este algoritmo permite usar la distancia de gower.

Tanto el proceso de simulación como la aplicación a datos reales muestran que CLARABD es consistente con PAM, las agrupaciones son similares desde distintos enfoques. En la simulación la consistencia se muestra a través del ratio dentro/entre (within/between). La aplicación usa tres elementos para mostrar la consistencia entre algoritmos.

En primer lugar, se presenta la matriz de confusión asumiendo la variable de incumplimiento como objetivo. Luego se muestran las coincidencias que existen entre los representantes (medoides) de los grupos obtenidos. Y, finalmente, se muestra un gráfico biplot donde los patrones lucen muy parecidos. Todas estas aproximaciones son evidencia de la coherencia del algoritmo propuesto: CLARABD.

Existen puntos específicos de CLARABD que pueden ser mejorados. Por ejemplo, por ahora, tanto la implementación de CLARA en R como la propuesta de CLARABD usa muestreo aleatorio simple para obtener las muestras del conjunto de objetos inicial. El diseño muestral podría ocasionar cambios radicales en las agrupaciones finales y responder de mejor manera a un problema específico. Por ejemplo, si se podría plantear un muestreo estratificado que recoja de mejor manera la probabilidad de selección y el peso que tiene cada objeto.

CLARABD es funcional y puede trabajar con más allá del umbral actualmente permitido por R (65536 objetos). No obstante, el tiempo de cómputo aún es un problema.

Tanto k-medoides como CLARA tienen su código fuente programado en el lenguaje C y R usa las funciones ejecutadas en C. Esta es una práctica común en R cuando se trata de procesos intensivos en cómputo.
Un trabajo derivado del presente podría implementar en C las rutinas más exigentes. En particular, la asignación de las observaciones no muestreadas a los medoides más cercanos es exigente computacionalmente.

Referencias

[1] Leonard Kaufman and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis, volume 344. John Wiley & Sons, 2009.
[2] Chris Fraley and Adrian E Raftery. How many clusters? which cluste- ring method? answers via model-based cluster analysis. The computer journal, 41(8):578–588, 1998.
[3] Pedro Galeano and Daniel Pen˜a. Data science, big data and statistics. TEST, pages 1–41, 2019.
[4] Surekha Borra, Rohit Thanki, and Nilanjan Dey. Satellite Image Analy- sis: Clustering and Classification. Springer, 2019.
[5] Brefeld Ulf, Jesse Davis, Jan Van Haaren, and Albrecht Zimmermann. Machine Learning and Data Mining for Sports Analytics. Springer, 2019.
[6] Gautam K Vallabha, James L McClelland, Ferran Pons, Janet F Werker, and Shigeaki Amano. Unsupervised learning of vowel categories from infant-directed speech. Proceedings of the National Academy of Sciences, 104(33):13273–13278, 2007.
[7] Feng Zhen, Xiao Qin, Xinyue Ye, Honghu Sun, and Zhaxi Luosang. Analyzing urban development patterns based on the flow analysis met- hod. Cities, 86:178–197, 2019.
[8] Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. cluster: Cluster Analysis Basics and Extensions, 2017. R pac- kage version 2.0.6 — For new features, see the 'Changelog' file (in the package source).
[9] Bater Makhabel. Learning data mining with R. Packt Publishing Ltd, 2015.
[10] R Core Team. R: A Language and Environment for Statistical Compu- ting. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL https://www.R-project.org/.
[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning,

ser, 2001.
[12] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning, volume 112. Springer, 2013.
[13] Alan Julian Izenman. Modern multivariate statistical techniques. Regression, classification and manifold learning, 2008.
[14] Salvador Godoy. Evaluacion de algoritmos de clasificacion basada en el modelo estructural de cubrimientos. PhD thesis, Instituto Polit´ecnico Nacional, México, 5 2006.
[15] Liset Bandomo Toledo. Procedimiento para evaluar el nivel de comple- jidad de los procesos de negocio a partir de su representacion grafica. PhD thesis, Universidad Central ?Marta Abreu? de Las Villas, 2014.
[16] A Riquenes-Fernandez and E Alba-Cabrera. Collective classification: An useful alternative for the classification of objects. In European Congress on Intelligent Techniques and Soft Computing EUFIT, volume 97, pages 1875–1879, 1997.
[17] Amparo Baillo Moreno and Aurea Gran´e Ch´avez. 100 problemas re- sueltos de estad´ıstica multivariante: (implementados en Matlab). Delta Publicaciones, Madrid, Espan˜a, 2008.
[18] J. MacQueen. Some methods for classification and analysis of multi- variate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pages
281–297, Berkeley, Calif., 1967. University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200512992.
[19] Brian D Ripley. Pattern recognition and neural networks. Cambridge university press, 2007.
[20] Rui Xu and Don Wunsch. Clustering, volume 10. John Wiley & Sons, 2008.
[21] Dan Toomey. R for Data Science. Packt Publishing Ltd, Unated King- dom, 1 edition, 2014.
[22] Alboukadel Kassambara. Statistical tools for high-throughput data analysis, 09 2018. URL http://www.sthda.com/english/.
[23] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.